

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Sistem

Menurut (Al-Bahra bin Ladjamudin, 2005:1-2), agar lebih mudah memahami apa dan bagaimana sistem itu akan digunakan dua pendekatan, yaitu:

a. Pendekatan Prosedur

Pemahaman sistem dengan pendekatan prosedur, yaitu suatu urutan kegiatan yang saling berhubungan, berkumpul bersama-sama untuk mencapai tujuan tertentu.

b. Pendekatan Komponen/Elemen

Pemahaman sistem dengan pendekatan komponen/elemen, yaitu kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan tertentu.

2.1.2 Karakteristik Sistem

Menurut (Al-Bahra bin Ladjamudin, 2005:4-5), sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik adalah sebagai berikut:

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai karakteristik dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

b. Batasan Sistem

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

c. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem sehingga harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem

Penghubung merupakan media yang menghubungkan antara subsistem dengan subsistem yang lainnya. Melalui penghubung ini kemungkinan sumber- sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem lainnya membentuk satu kesatuan.

e. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal maintenance *input* yaitu energi yang dimasukkan agar sistem tersebut dapat berjalan. Sinyal *input* adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

f. Keluaran Sistem

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

g. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, jika sistem tidak mempunyai sasaran maka sistem tidak akan ada.

2.1.3 Pengertian Sistem Informasi

Sistem Informasi dapat didefinisikan sebagai berikut:

- a. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
- b. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan organisasi.
- c. Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Al Bahra bin Ladjamudin, 2005:13-14)

2.2 Pengertian Sistem Pendukung Keputusan (SPK)

Menurut Kusrini (2007), SPK merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, pemanipulasian data. Sistem itu digunakan untuk membantu pengambil keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana seharusnya keputusan tersebut dibuat.

SPK tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. Tujuan dari SPK adalah:

1. Membantu manajer dalam pengambilan keputusan atas masalah semi-terstruktur.

2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang diambil manajer lebih dari pada perbaikan efisiensi.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya rendah.
5. Peningkatan produktivitas. Membangun satu kelompok pengambil keputusan, terutama para pakar.
6. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat. Sebagai contoh, semakin banyak data yang diakses, makin banyak pula alternatif yang bisa dievaluasi.
7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan pengambilan keputusan menjadi sulit.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan menurut Simon (1977), otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi.

2.3 Fuzzy Multiple Attribute Decision Making (FMADM)

Fuzzy Multiple Attribute Decision Making (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan (Sri Kusumadewi, 2006:135).

Pada dasarnya, ada 3 (tiga) pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif dan obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan. Ada beberapa metode yang dapat

digunakan untuk menyelesaikan masalah FMADM. antara lain (Sri Kusumadewi, 2006:74):

1. *Simple Additive Weighting Method (SAW)*
2. *Weighted Product (WP)*
3. *ELECTRE*
4. *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)*
5. *Analytic Hierarchy Process (AHP)*

2.4 Metode Simple Additive Weighting Method (SAW)

Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,...,m$ dan $j=1,2,...,n$.

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai:

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Keterangan :

V_i = rangking untuk setiap *alternative*

W_j = nilai bobot untuk setiap kriteria

r_{ij} = nilai rating kinerja ternormalisasi

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih (Sri Kusumadewi, 2006:74).

2.5 Algoritma Penyelesaian

Dalam penulisan laporan ini menggunakan FMADM dengan metode SAW, langkahnya adalah sebagai berikut:

1. Memberikan nilai setiap alternatif (A_i) pada setiap kriteria (C_j) yang sudah ditentukan, dimana nilai tersebut di peroleh berdasarkan nilai crisp; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$.
2. Memberikan nilai bobot (W) yang juga didapatkan berdasarkan nilai crisp.
3. Melakukan normalisasi *matriks* dengan cara menghitung nilai rating kinerja ternormalisasi (r_{ij}) dari alternatif A_i pada atribut C_j berdasarkan persamaan yang disesuaikan dengan jenis atribut (keuntungan/*benefit*=MAKSIMUM atribut biaya/*cost*=MINIMUM). Apabila berupa atribut keuntungan maka nilai *crisp* (X_{ij}) dari setiap kolom atribut dibagi dengan nilai crisp *MAX* (*MAX* X_{ij}) dari tiap kolom, sedangkan untuk atribut biaya, nilai crisp *MIN* (*MIN* X_{ij}) dari tiap kolom atribut dibagi dengan nilai crisp (X_{ij}) setiap kolom.
4. Melakukan proses perankingan dengan cara mengalikan *matriks* ternormalisasi (R) dengan nilai bobot (W).
5. Menentukan nilai preferensi untuk setiap alternatif (V_i) dengan cara menjumlahkan hasil kali antara *matriks* ternormalisasi (R) dengan nilai bobot (W). Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih (Kusumadewi, 2007).

2.6 Pengertian Beasiswa

Pengertian beasiswa seperti yang dikutip dari www.wikipedia.org adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan.

Pemberian beasiswa dapat dikategorikan pada pemberian cuma-cuma ataupun pemberian dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan. Lama ikatan dinas ini berbeda-beda, tergantung pada lembaga yang memberikan beasiswa tersebut.

2.6.1 Maksud dan Tujuan Pemberian Beasiswa

1. Maksud dari pemberian beasiswa adalah untuk memberikan bantuan subsidi biaya pendidikan dalam bentuk uang kepada siswa tidak/kurang mampu.
2. Tujuan dari pemberian beasiswa antara lain adalah :
 - a. Membantu siswa tidak/kurang mampu untuk memenuhi kebutuhan sekolah/madrasah.
 - b. Mencegah siswa tidak/kurang mampu dari kemungkinan putus sekolah akibat kesulitan ekonomi.
 - c. Memberi peluang dan kesempatan yang lebih besar kepada siswa tidak/kurang mampu untuk menyelesaikan pendidikan sampai dengan jenjang pendidikan Sekolah Menengah Atas/Madrasah Aliyah/Sekolah Menengah Kejuruan. (Perbup No 27, 2010)

2.7 Pemodelan Objek

Pemodelan objek merupakan suatu metode untuk menggambarkan struktur sistem yang memperlihatkan semua objek yang ada pada sistem. Model yang digunakan untuk menyederhanakan cara mengkomunikasikan proses-proses tersebut terbagi atas (Suhendar,2002:12-13):

a. Objek

Objek (*object*) adalah Benda, secara fisik atau konseptual, yang dapat kita temui disekeliling kita.

b. Kelas

Kelas (*Class*) adalah Definisi secara umum (pola, *template* atau cetak biru) untuk menghimpun objek sejenis

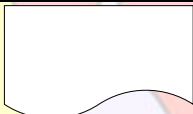


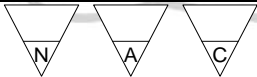
2.8 Pemodelan Proses

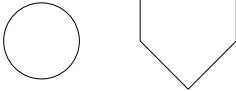
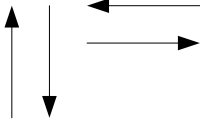

Pemodelan proses digunakan untuk mengilustrasikan aktivitas-aktivitas yang dilakukan dan bagaimana data berpindah diantara aktivitas-aktivitas tersebut. Cara untuk merepresentasikan proses model dengan menggunakan UML (*Unified Modeling Language*). UML merupakan sistem arsitektur yang bekerja dengan OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* yang terdapat dalam sistem *software*.

2.9 Flow Of Documents

Bagan alir dokumen (dokumen *flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan didalam bagan alir sistem

Tabel 2.1: Simbol-simbol *Flow of Document*

Simbol	Keterangan
	Dokumen, menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer
	<i>Keyboard</i> , menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i>
	Kegiatan manual, menunjukkan pekerjaan manual
	Simpanan <i>offline</i> , <i>file non-komputer</i> yang diarsip menurut angka (<i>numerical</i>), huruf (<i>alphabetical</i>), tanggal (<i>cronological</i>)

	Penghubung, menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain
	Garis alir, menunjukkan arus dari proses
	Proses, menunjukkan kegiatan proses dari operasi program komputer

(Sumber: Jogiyanto, HM, *Analisis dan Desain*, 2005)

2.9.1 UML (*Unified Modelling Language*)

Menurut Suhendar (2002:26) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifacts* dari sistem *software*, untuk memodelkan bisnis, dan sistem *nonsoftware* lainnya. UML merupakan suatu kumpulan teknik terbaik yang terbukti sukses dalam memodelkan sistem yang besar dan kompleks.

2.9.2 Tujuan UML

Tujuan utama UML diantaranya untuk:

- Memberikan model yang siap pakai bahasa pemodelan visual yang ekspresif untuk mengemban saling menukar model dengan mudah dan dimengerti secara visual.
- Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

(Suhendar, 2002:30)

2.9.3 *Artifact* UML



Menurut Suhendar (2002:26) menyatakan bahwa *artifact* adalah sepotong informasi yang digunakan atau dihasilkan dalam proses rekayasa *software* dapat berupa model deskripsi, atau *software*. Pemodelan proses digunakan untuk mengilustrasikan aktivitas-aktivitas yang dilakukan dan bagaimana data berpindah diantara aktivitas-aktivitas tersebut. Cara untuk merepresentasikan proses model

dengan menggunakan UML (*Unified Modeling Language*). UML merupakan sistem arsitektur yang bekerja dengan OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi dan mendokumentasikan *artifact* yang terdapat dalam sistem *software*. Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:

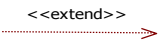

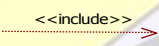
a. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* di gunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa A.S-M.Shalahuddin, 2011:137)

Tabel 2.2: *Use Case Diagram*

No	Nama	Gambar	Fungsi
1	<i>Use case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	<i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	<i>Association</i>		Komunikasi antara aktor dengan <i>use case</i> yang berpartisipasi pada <i>use case</i>

Lanjutan Tabel 2.2

			atau <i>use case</i> memiliki interaksi dengan aktor.
4	<i>Extend</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5	<i>Generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

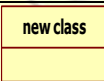


Sumber: (Rosa A.S-M. Shalahuddin, 2011:137)





b. *Class Diagram*

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa A.S-M.Shalahuddin, 2011:126).

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.3: *Class Diagram*

No	Nama	Gambar	Fungsi
1	<i>Class</i>		Kelas pada struktur sistem.
2	<i>Interface</i>		Kumpulan operasi tanpa implementasi dari suatu <i>class</i> .
3	<i>Association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .



4	<i>Directed Association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	<i>Generalisasi</i>		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6	<i>Dependency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas.
7	<i>Aggregation</i>		Semua bagian (<i>whole-part</i>)

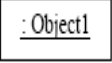

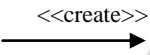
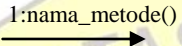

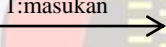
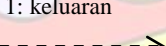
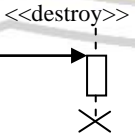
Sumber: (Rosa A.S-M. Shalahuddin, 2011:126)

c. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa A.S-M. Shalahuddin, 2011:145).

Tabel 2.4: *Sequence Diagram*

No	Nama	Gambar	Fungsi
1	<i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.
2	Garis hidup/ <i>Lifeline</i>		Menyatakan kehidupan suatu objek






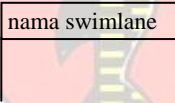
3	<i>Object</i>		Menyatakan objek yang berinteraksi pesan
4	Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
5	Pesan tipe <i>create</i>		Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6	Pesan tipe <i>call</i>	 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada <i>diagram</i> kelas sesuai dengan kelas objek yang berinteraksi.
Lanjutan Tabel 2.4			
Bersambung			
7	Pesan tipe <i>send</i>		Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8	Pesan tipe <i>return</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	Pesan tipe <i>destroy</i>		Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang di akhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: (Rosa A.S-M. Shalahuddin, 2011:145)

d. *Activity Diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Rosa A.S-M. Shalahuddin, 2011:141)

Tabel 2.5: *Activity Diagram*


No	Nama	Gambar	Fungsi
1	Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan/ Lanjutan Tabel 2.5		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/ Bersambung <i>join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi




Sumber: (Rosa A.S-M. Shalahuddin, 2011:141)

e. *Statechart Diagram*

Statechart diagram, atau yang biasa juga disebut *state diagram* memperlihatkan berbagai *state* (keadaan sesaat) yang dilalui sebuah objek, dan kejadian-kejadian yang menyebabkan sebuah transisi dan satu *state* ke *state* lainnya, dan aksi yang mengakibatkan suatu perubahan *state*. (Suhendar, 2002:54).

Tabel 2.6: *Statechart Diagram*

Gambar	Nama	Keterangan
	<i>Start State</i>	Menggambarkan permulaan suatu proses

	<i>Transition</i>	Hubungan antara dua <i>state</i> yang menunjukkan kapan sebuah objek dapat bergerak kepada <i>state</i> lainnya.
	<i>End State</i>	Menggambarkan akhir suatu proses
	<i>State</i>	Menjelaskan keadaan tertentu suatu objek selama memenuhi suatu syarat atau kondisi tertentu.

Sumber: (Suhendar, A, 2002:54)

2.10 Pengertian Basis Data

Menurut (Rosa A.S-M. Shalahuddin, 2011:44) Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada dasarnya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data dalam system informasi meliputi:

1. Memasukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah tersimpan

2.11 Diagram Hubungan Entitas (*Entity Relation Diagram*)

Menurut Al-Bahra bin Ladjamudin (2005:142) *Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Jadi, jelaslah bahwa ERD ini berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan relationship data. Elemen-elemen dalam *Diagram Hubungan Entitas* meliputi :

a. *Entity*

Pada E-R *Diagram*, *Entity* dapat digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam

empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

b. Relationship

Pada E-R *Diagram*, *Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau pasif).

Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

1. *Relationship Degree*

Relationship Degree atau Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam suatu *Relationship*.

a. *Unary Degree* (Derajat Satu)



Gambar 2.1: *Unary Degree*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

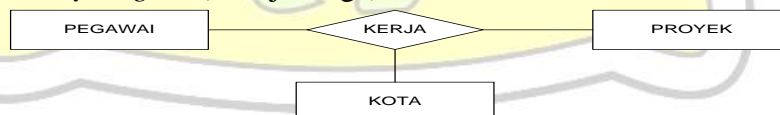
b. *Binary Degree* (Derajat Dua)



Gambar 2.2: *Binary Degree*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

c. *Ternary Degree* (Derajat Tiga)



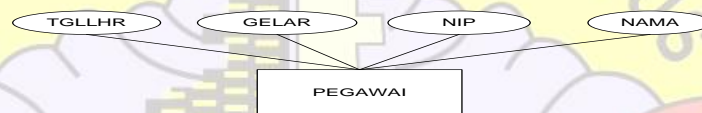
Gambar 2.3: *Ternary Degree*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

2. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *Relationship*. Atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *Relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *Relationship*. Atribut *Value* atau Atribut Nilai adalah suatu *occurence* tertentu dari sebuah atribut di dalam suatu *entity* atau *Relationship*. Ada dua jenis-jenis atribut :

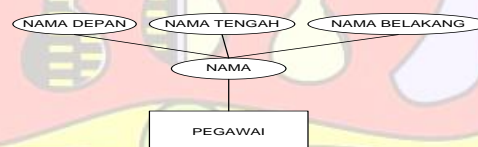
1. *Key* adalah atribut yang digunakan untuk menentukan suatu *entity* secara unik.
2. Atribut *Simple* adalah atribut yang bernilai tunggal.
3. Atribut *Multivalued* adalah atribut yang memiliki sekelompok nilai untuk setiap *instan entity*.



Gambar 2.4: *Atribut Multivalued*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

4. Atribut *Composite* adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.



Gambar 2.5: *Atribut Composite*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

3. Kardinalitas

Kardinalitas Relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antara entitas tersebut, Kardinalitas Relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu:

a. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua dan sebaliknya. Yang berarti setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B, dan begitu juga sebaliknya setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.

Contoh : Adanya relasi antara entitas Dosen dengan entitas Jurusan. Relasinya diberi nama 'Kepala'. Pada relasi ini, setiap dosen paling banyak mengepalai satu jurusan (walaupun memang tidak semua dosen yang menjadi ketua jurusan). Dan setiap jurusan dikepalai oleh paling banyak satu orang dosen.



Gambar 2.6: *Diagram Kardinalitas One to One*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

b. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. *One to Many*

Satu tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas B, berhubungan dengan paling banyak satu tupel pada entitas A.

Contoh : Adanya relasi antara entitas Dosen dengan entitas Kuliah. Relasinya diberi nama 'Ajar'. Setiap dosen dapat mengajar lebih

dari satu mata kuliah, sedang setiap mata kuliah hanya oleh paling banyak satu orang dosen.



Gambar 2.7: *Diagram Kardinalitas One to Many*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

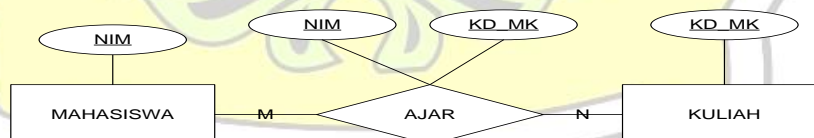
d. *Many to One*

Setiap tupel pada entitas A dapat berhubungan dengan paling banyak satu tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B.

e. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika setiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua. Yang berarti setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, dan demikian juga sebaliknya, dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A.

Contoh : Adanya relasi entitas Mahasiswa dengan entitas Kuliah. Relasinya diberi nama 'Belajar'. Setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap mata kuliah dapat dipelajari oleh lebih dari satu orang mahasiswa.

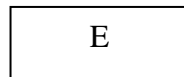


Gambar 2.8: *Diagram Kardinalitas Many to Many*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

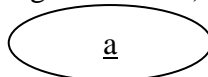
Notasi-notasi simbolik di dalam *Diagram E-R*:

1. Persegi panjang, menyatakan Himpunan Entitas/entitas.

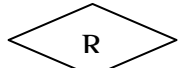


Himpunan Entitas/entitas E

2. Lingkaran/*Elip*, menyatakan Atribut (Atribut yang berfungsi sebagai *key* digaris bawah). Atribut *a* sebagai *key*



3. Belah ketupat menyatakan Himpunan Relasi/relasi.



Himpunan Relasi/relasi R

4. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.

Link

Langkah–langkah teknis untuk menghasilkan *Entity Relationship Diagram* (ERD):

- a. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non key*).
- b. Mengidentifikasi dan menetapkan seluruh entitas yang akan terlibat.
- c. Menentukan atribut-atribut *key* (*primary key*) dari masing-masing entitas.
- d. Mengidentifikasi dan menetapkan seluruh relasi diantara entitas-entitas yang ada beserta *foreign-key*-nya (jika terjadi kardinalitas relasi *One to Many* atau *Many to Many*)
- e. Menentukan derajat/kardinalitas relasi untuk setiap relasi.

2.12 Transformasi dari ERD ke *Database Relasional*

1. Setiap tipe *Entity* dibuat suatu relasi yang memuat semua atribut *simple*, sedangkan untuk atribut *composite* hanya dimuat komponen-komponennya saja.

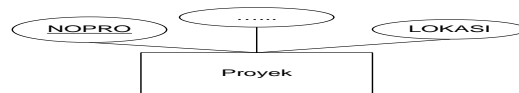


PEGAWAI (NOPEG, ALM1, KDPOS,)

Gambar 2.9: Transformasi dari ERD ke *Database*

Sumber: Al-Bahra bin Ladjamudin (2005:164)

2. Setiap relasi yang mempunyai atribut multivalued, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key dari relasi tersebut dengan atribut multivalued.



LOKPR(NOPRO, LOKASI)

Gambar 2.10: Transformasi dari ERD ke *Datasase*

Sumber: Al-Bahra bin Ladjamudin (2005:165)

3. Setiap *Unary Relationship* 1:N, pada relasi perlu ditambahkan suatu *foreign key* yang menunjuk ke nilai *primary keynya*.

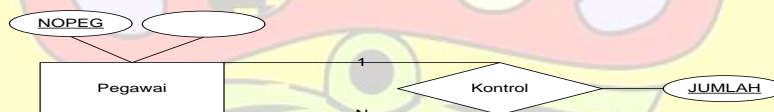


PEGAWAI (NOPEG,, SUPERVISOR-ID)

Gambar 2.11: Transformasi dari ERD ke *Database*

Sumber: Al-Bahra bin Ladjamudin (2005:166)

4. Setiap *Unary Relationship* M:N, buatlah relasi baru dimana *primary keynya* merupakan gabungan dari dua atribut dimana keduanya menunjuk ke *primary key* relasi awal dengan penamaan yang berbeda.



KOMBAR (NOBAR, NOKOMP, JUMLAH)

Gambar 2.12: Transformasi dari ERD ke *Database*

Sumber: Al-Bahra bin Ladjamudin (2005:167)

5. Setiap *Binary Relationship* 1:1, dimana *Participation Constraint* keduanya total, buatlah suatu relasi gabungan dimana *Primary Keynya* dapat dipilih salah satu.



PEGAWAI (NOPEG, ... , NOPRO, ...).

Gambar 2.13: Transformasi dari ERD ke Database

Sumber: Al-Bahra bin Ladjamudin (2005:168)

6. Setiap *Binary Relationship* 1:1 dan salah satu *Participation Constraint*nya Total, maka *Primary Key* pada relasi yang *Participation Constraint*nya Partial menjadi *Foreign Key* pada relasi yang lainnya.



BAGIAN (NOBAG, ... , MANAGER)

Gambar 2.14: Transformasi dari ERD ke Database

Sumber: Al-Bahra bin Ladjamudin (2005:169)

7. Setiap *Binary Relationship* 1:1, dimana kedua *Participation Constraint*nya partial, maka selain kedua relasi perlu dibuat relasi baru yang berisi *Primary Key* gabungan dari *Primary Key* kedua tipe *Entity* yang berelasi.



PEKERJAAN (NOPEG, NOPRO, ...)

Gambar 2.15: Transformasi dari ERD ke Database

Sumber: Al-Bahra bin Ladjamudin (2005:170)

8. Setiap *Binary Relationship* 1:N, dimana tipe *Entity* yang bersisi N mempunyai *Participation Constraint* Total, maka *Primary Key* pada relasi yang bersisi 1 dijadikan *Foreign Key* pada relasi yang bersisi N.



PROYEK (NOPRO, ... , NOBAG)

Gambar 2.16: Transformasi dari ERD ke Database

Sumber: Al-Bahra bin Ladjamudin (2005:171)

9. Setiap *Binary Relationship* 1:N, dimana tipe *Entity* yang bersisi N mempunyai *Participation Constraint* partial, buatlah relasi baru dimana

*Primary Key*nya merupakan gabungan dari *Primary Key* kedua tipe *Entity* yang berelasi.



PEKERJAAN (NOPEG, NOPRO,)

Gambar 2.17: Transformasi dari ERD ke *Datasase*

Sumber: Al-Bahra bin Ladjamudin (2005:172)

10. Setiap *Binary Relationship* M:N, buatlah relasi baru dimana *Primary Key* nya merupakan gabungan dari *Primary Key* kedua tipe *Entity* yang berelasi.

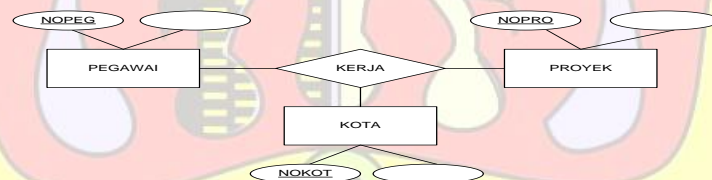


PEKERJAAN (NOPEG, NOPRO,)

Gambar 2.18: Transformasi dari ERD ke *Database*

Sumber: Al-Bahra bin Ladjamudin (2005:173)

11. Setiap *Ternary Relationship*, buatlah relasi baru dimana *Primary Key*nya merupakan gabungan dari *Primary Key* ketiga tipe *Entity* yang berelasi.

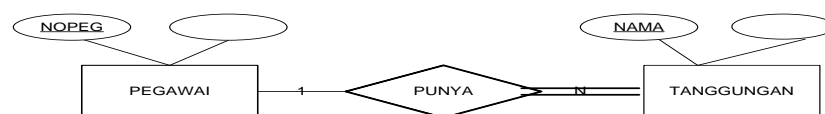


PEKERJAAN (NOPEG, NOPRO , NOKOT)

Gambar 2.19: Transformasi dari ERD ke *Database*

Sumber: Al-Bahra bin Ladjamudin (2005:174)

12. Setiap tipe *Weak Entity*, dibuat suatu relasi yang memuat semua atributnya dimana *Primary Key*nya adalah gabungan dari *Partial Key* dan *Primary Key* dari relasi induknya (*identifying owner*).



TANGGUNGAN (NOPEG, NAMA,)

Gambar 2.20: Transformasi dari ERD ke *Database*

Sumber: Al-Bahra bin Ladjamudin (2005:175)

